Towards unified QoS/SLA ontologies

Glen Dobson
Computing Department
Lancaster University
g.dobson@comp.lancs.ac.uk

Alfonso Sánchez-Macián
Dept. Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
aasmp@dit.upm.es

Abstract

This paper reports on work in progress to produce a unified Quality of Service (QoS) ontology This ontology must be applicable to the main scenarios currently identified such as QoS-based Web Services selection, QoS monitoring and QoS adaptation. An evaluation of existing research in the field of QoS and Service Level Agreement (SLA) ontologies is presented. Many of the authors of these works are involved in the initiative. The aim of this evaluation is to identify the weaknesses and strengths of existing ontologies in order to decide which parts of each should form the basis of a unified ontology. Current progress in the ontology engineering process is also presented.

1. Introduction

The need for a Quality of Service (QoS) ontology in a web service selection scenario has been widely discussed, for instance [4] or [6]. QoS semantics allows the user (and its applications) to define service quality requirements. Providers publish the QoS of the services being offered using this ontology. Finally, a broker is able to perform an automatic selection of the best service offer (in terms of quality and price) among those made available by providers.

Many efforts have been undertaken concerning the representation of quality of service concepts. Some of them are oriented to define specific languages (such as WSLA [16] or WS-Agreement [17]) to present agreements including quality of service. Other works describe quality concepts in standards (e.g.: ITU-T E-800 [1]), but are not published in a format that computers can understand. Finally, some of them are centered on the definition of a QoS ontology to provide a machine-understandable semantic representation of QoS information.

This paper evaluates the QoS ontology research work currently available, and presents an initiative to create a unified ontology. To do this, first section introduces some scenarios of application for a QoS ontology. Next, existing ontologies are studied and evaluated. Finally, the current state of the initiative is presented

2. Application scenarios

QoS ontologies can be utilized in various ways in order to address different issues. This section presents some of the most relevant scenarios of application.

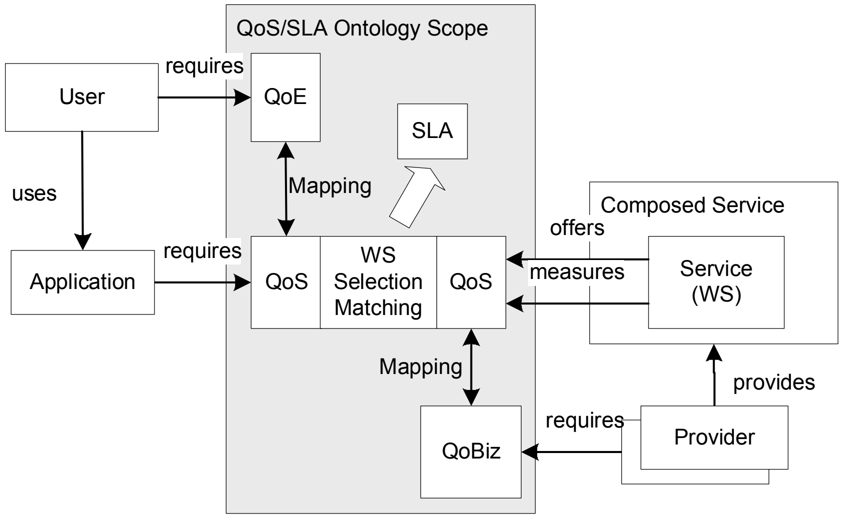

Figure 1. QoS-based WS selection

The first scenario is that which was already mentioned in the introduction: QoS-based service selection. In a B2B or B2C environment, different offers can provide similar service functionality. A customer requires tools to decide which of those services fits its needs best. The use of ontologies and rules makes it possible to perform a matching between user QoS requirements and service offers in real time. The selection can be supported by a broker or a reasoner.

A complementary approach consists of comparing the user requirements not only to the information published by the provider, but also to the information measured for the service. A more complex situation (Figure 1) appears when taking into account the different QoS facets [3]: the technical QoS, the user perception of quality or Quality of Experience (QoE) and the Quality of Business (QoBiz). The definition of additional rules and mathematical semantics to map and translate those quality levels (as in [13]) helps on supporting the whole business process.

A second application topic is the QoS monitoring and detection of SLA violations. In this case, measurement is made to detect the real behavior of the service. Then a measured profile is constructed to be compared to the agreements included in the SLA. In this case, measured values replace user requirements in the comparison made by the broker.

Finally, it can also be used for QoS adaptation. Service quality information is measured in real time. Some controlled parameters of the service can be modifying to react to a possible degradation of quality, for instance reducing the number of connections or replicating some elements of the service infrastructure.

3. Evaluation of existing ontologies

There are many research efforts and papers in the existing literature for developing and using a QoS ontology. In this section the most significant are analyzed.

Tosic *et al.* [4] enumerate some requirements for a QoS ontology, including a set of necessary ontologies: measurement units, currency units, measured properties, measurement methods and metrics. These are general requirements that must be completed with other related to the scenarios of application.

FIPA [5] defines a low-level ontology oriented to network aspects. Although it can serve for agent communication in a network scenario, it is too specific to be applicable to the required goals. In particular, it fails to define a substantial upper ontology meaning that it is difficult to extend to other aspects of QoS. The ontology has also been applied only in a FIPA architecture and therefore is not, as it stands, directly applicable in a web services environment.

Tian *et al* [6] defines a WS-QoS framework that includes a QoS ontology. This ontology is used to perform dynamic selection of Web services based on service or network requirements. There is no difference in this ontology between a characteristic (what is going to be measured) and the method used to measure. So, to compare offers, each service should provide values of the characteristic measured using the same metric. Looking more closely, the WS-QoS ontology is in fact an XML language, with no formally defined semantics. It therefore lacks the advantages for machine reasoning that, e.g. an OWL [20] ontology has. The semantics would have to reside in the application code and therefore be open to misinterpretation by developers. The lack of formally stated semantics makes this problem even worse.

In contrast, DAML-QoS [7], is an OWL ontology for QoS in web services. In fact the ontology was originally encoded using DAML+OIL – the predecessor to OWL, but is now evolving to use OWL, and changing in name appropriately to OWL-QoS. DAML-QoS has many advantages, including its links to the OWL-S [18] ontology, which can be used for describing web services. As in [6], metrics are used directly without relating them to what attribute they measure. As well as QoS description, DAML-QoS supports concepts such as QoS adverts and inquiries, and the semantics of matching are formally described. Unfortunately the approach used is flawed in that it uses cardinality constraints to express bounds upon QoS properties. As the term cardinality suggests, this is actually a misuse of this OWL construct. A cardinality constraint puts constraints on the number of values a property can take, not on the values themselves. Even if the approach taken was valid, it also carries the limitation that it can only express bounds as positive integers (e.g. there is no simple way to say "availability> 0.999").

Maximilien and Singh [8] present a framework, which uses an ontology to support dynamic web services selection. Despite its promise, this ontology lacks both an openly available implementation and links to OWL-S.

MOQ [9] makes a good attempt at defining the semantics of QoS. It correctly notes that QoS requirements may be composite – but fails to suggest a means to allow logical combinations of requirements, only stating that if *all* sub-requirements are met then the composite is always satisfied. Unfortunately, depending upon one's interpretation, this may not be true. For instance consider what happens if one allows negation when combining requirements – e.g. R1 NAND R2 is specifically met only when requirements R1 and R2 are *not* both simultaneously met. MOQ decomposes the concept of measurements further than most onther ontologies, by allowing a sampling plan to be specified. MOQ also correctly identifies the importance of traceability (which might perhaps also be termed accountability or provenance), which is not dealt with elsewhere. Unfortunately, the major drawback to MOQ is that it does not in itself seem to present an ontology, but only talks about the semantics of QoS ontologies in general. It therefore has no use as a vocabulary and taxonomy of QoS terms and fails to address all of the issues that complete ontologies do, despite filling in gaps which are missing from some of these.

In [10], an attempt is made to define the relevant concepts of quality of service. However, this work does not go on to define a full ontology, and it lacks many concepts which more complete ontologies already cover. Some supporting concepts are missing, for instance those related to units. It does not define agreement concepts (offers or requirements) either. Overall, the concepts lack definition (beyond giving them a name and some relationships) and the semantics of, e.g. matching/discovery are not discussed.

QoSOnt [11] has much in common with other OWL ontologies for web services (e.g. [7]). It contains links to OWL-S and concentrates upon the definition of metrics and on requirements matching. Unfortunately, despite identifying the correct semantics for matching (applying universal quantification over a metric, with a user-defined datatype as the range and then using OWL subclassing to find all metrics matching this condition) this has never been demonstrated using QoSOnt and an associated XML language has been used, losing many of the advantages of OWL. This is mainly due to the poor datatyping support in OWL and therefore in OWL tools. In the new version of OWL (version 1.1) currently in draft, the datatype problem is solved and therefore the semantics suggested could perhaps be implemented more easily and eventually supported in a standard way in other OWL tools.

As well as pointing the direction to the correct semantics for matchmaking QoSOnt also correctly identifies that the value of a metric is only relevant in the correct scope (e.g. network latency applies to a particular network route, time-to-complete applies to a particular service operation, etc.) and that metrics have a "direction" e.g. higher is better. Initial attempts at representing how metrics combine when services are composed have also been made.

[12] is another QoS ontology for web services with an OWL implementation. Unlike DAML-QoS and QoSOnt, however, it does not define links to OWL-S. Its chief differences are in the addition of the QoSImpact, Node and Nature concepts. The first of these indicates how a particular QoS parameter affects overall quality (ie.g. decreased latency is better quality), the second allows the relationship between quality and the actual server/clients involved to be represented, the latter denotes that a parameter is either static or dynamic (i.e. can change at runtime).

Oldham et al [14] takes a slightly different approach to that taken by the other work discussed here, attempting to add semantics to an essentially syntactic SLA technology. In this work they have attempted to demonstrate how to add semantics to WS-Agreement [17] using OWL and SWRL [19] rules. This is achieved by adding tags to WS-Agreement to link it to the relevant semantics. This solution improves greatly on situations where a pure syntactic match is difficult due to terminological ambiguity. The approach demonstrated is valuable in that it suggests how to integrate with legacy languages as well as the concepts it suggests are present in WS-Agreement which are missing from other ontologies., e.g. "business values".

4. Current status of the process

An increasing number of members (currently 20) from different countries and affiliations are joining the mailing list [15] that supports this project.

The project is defining two ontologies simultaneously: a QoS ontology and a SLA ontology. The former includes those terms strictly related to quality of service, such as Metric, Characteristic or Value. The SLA ontology is supported by the QoS ontology and represents all the terms that are necessary to define a SLA, define offers and requirements and perform dynamic selection between different service offers.

A well-known development methodology [2] was selected to create the top-level QoS and SLA ontologies. This process is composed of seven steps that allow the generation of an initial version of the ontology: 1) determine the domain and scope of the ontology, 2) consider reusing existing ontologies, 3) enumerate important terms in the ontology, 4) define the classes and the class hierarchy, 5) define the properties of classes-slots, 6) define the facets of the slots and 7) create instances.

The first version can be refined in an iterative way (repeating the seven steps). The top-level ontology can be mapped to more specific concepts in latter iterations.

The first three steps (in a first iteration) have been already completed:

- The scope (QoS and SLA) and application scenarios (previously commented) have been defined. A list of competency questions that should be answered using the ontology was also created, e.g.: which services offer a QoS good enough for a user/application request?
- The existing ontologies have already been studied and it was decided to take parts of all of them, but not using a specific one as a basis.
- A list of important terms (and also relationships) has been enumerated. They have been separated in QoS and SLA concepts to be allocated in the correspondent ontologies.
- A set of requirements have been enumerated.

4.1 General requirements of the ontology

One of the initial tasks performed by the researchers at the project has been the identification of the requirements that have to be fulfilled when developing a QoS ontology. They have been divided in three groups: general, QoS-related and SLA-related.

The general requirements are defined below:

- The Ontology must define all the top-level QoS concepts necessary to be used in a real application.
- The Ontology must be applicable to different scenarios such as Web services selection, QoS monitoring or SLA violation detection.
- The Ontology must include well-defined semantics for the "advertisement-request" matching, and specification validation.
- The Ontology should align itself with other established ontologies such as OWL-Time and OWL-S where relevant.
- The Ontology should be supported by an ontology of units and dimensions with rules for unit conversion when different units are met.
- The Ontology may allow Description Logic reasoning.
- The Ontology may be supported by an OOP design. This can align the service QoS definition to its corresponding coding software's OOP design to achieve better reuse.
- The Ontology may include auto-validation to check whether the advertisements and SLAs are consistent and valid.
- The Ontology should define the concepts using names widely accepted (standards, papers).
- The Ontology should support composability.

These requirements should be considered as goals that outputs of the initiative can be measured and compared against.

4.2 QoS concepts and requirements

The main concepts belonging QoS field have been selected from the definition made in previous works (e.g. [7], [8], [10] or [11]) and extended by the result of internal discussion. They are presented next:

- Characteristic or Attribute (e.g.: Delay): represents any attribute of a service related to its quality.
- Value or Measurement: the result of measuring a characteristic using a metric.
- Metric: a method to measure a characteristic.
- Concepts to characterize the measurement (when/where/how information) and the metrics (composability).
- Function: to relate metrics. Functions can be represented as concepts, rules in a rule language or both.
- Level (where attribute is located)
- QoS profile: a set of metric values for a service. Two more specific concepts are included: Reference Level and Measured Level.
- Provider, Customer and Third Party, derived from Actor concept.
- ValueType: can be numeric (real, integer) or literal (string)

There are some other necessary concepts provided by supporting ontologies: Unit (e.g.: kbps), Dimension of the unit and System of Units from an ontology of units; Service and Actor from a service ontology (e.g.: OWL-S).

These concepts must accomplish the following requirements that can be translated into properties.

- A service category must be related to a set of Characteristics that affect its quality.
- The quality of a service must be evaluated using a set of Metrics.
- A metric must define a method to obtain Measurements of a Characteristic.
- A value must be categorisable along a number of orthogonal parameters, at least including network route (if applicable), time of measurement, the QoS attribute (or characteristic) being measured, and service element being measured (i.e. is it a particular operation that is being measured or some measure for the complete service QoS?).
- A measurement must carry information about when (e.g. time of measurement, sample frequency, interval), where (locations) and how (Function) they were measured.
- A Measurement must define the Unit (or Unitless) in which its value is stated.
- A Characteristic should define which level it belongs to (e.g. architectural, functional, channel, and user)
- If a Characteristic is measured in units, it must be related to a unit dimension.
- A Function must define how to calculate a value from other values of the same characteristic (e.g. mean) or different characteristics (e.g.: success login/total tries). These functions can be implemented as rules.
- A measurement must define which service offer it is evaluating.
- A Measurement or Value should belong to a specific ValueType. For Literal ValueTypes, there must be a function (can be represented as a SWRL rule) specifying if one value is greater than another one.
- A Unit must be related to a Unit Dimension.
- A Unit must belong to a "System of Units" concept.

- A Unit can be a Basic Unit, Unit Multiple, Non-Basic Unit.

4.3 SLA concepts and requirements

To cope with a service selection scenario with service offers and user requirements, it is necessary to define additional terms. Some of the works already discussed (e.g.: [7], [14]) introduce some relevant concepts. These have been added to in the project mailing list:

- Service Level Agreement. Quality is only a part of a SLA.
- Three specific concepts derived from QoS profile Guarantee/Agreement, Advertisement and User request.
- Qualifying conditions to specify conditions for a SLA to be valid.
- Service Level Objective (SLO) (also called Assertion or Constraint) with two derived concepts: Capability (Offer) and Requirement. The fulfillment of the objectives can be verified using rules.
- A concept to indicate which of the actors in the agreement is required to fulfill a SLA condition.
- Quality assessment concepts. Assessment (Match/Mismatch).
- Business Value (Penalty, Reward, Importance)
- Assertion specification concepts (e.g.: predicate, comparison operator, other)
- Cost Model information.

Some relationships between these concepts have been expressed as requirements for a SLA ontology:

- A SLA must include the service provider, service requester, and measurement partners.
- A SLA must define when to start the SLA, what the lease is, the repetition interval and duration of the service.
- A SLA must define Business values (penalties, rewards and importance), e.g. the fine when certain target fails to obey the SLA.
- A SLA must define the price of the service (the cost model).
- A SLA may include a set of Qualifying conditions.
- A SLA must define a Guarantee (quality Agreement) between the provider and the customer.
- A QoS Profile must define a set of SLOs. There are several types of profiles: Advertisement (or Provider offer), Agreement(or Guarantee), User Request, Reference Level and Measured Level.
- A SLO (Requirement or Capability assertion) must define a Value (e.g.: threshold, range) for a Characteristic of a Service using a Metric.
- An Assessment (Match/Mismatch) must specify a relationship between two Profiles/Service Levels.
- A Match between profiles (e.g. an advertisement and a request) should specify the grade of matching in order to compare with other Matches.

5. Conclusion and further work

In this paper, we have evaluated existing QoS ontologies, explaining deficiencies and possible improvements to be made to each of them. We also present an initiative being undertaken to create a unified QoS ontology and the initial results. This project includes developers from all over the world joining their effort to generate a common ontology and create a standardized method to represent quality of service and SLAs.

Future work includes the generation of documentation defining the agreed ontology and examples of its application. It will also be submitted to an international organization (W3C) to publicize it and get feedback from users.

6. Acknowledgment

The authors would like to thank Zhou Chen, Kunal Verma, Kyriakos Kritikos, and all other people in the OWL-QoS mailing list who have contributed to the definition of the requirements and concepts presented in this paper.

Glen Dobson would also like to acknowledge the Service Centric System Engineering (SeCSE) EU Integrated Project 511680, which has funded his work in this area.

Work from Alfonso Sánchez-Macián is supported by the Spanish National Plan of Research, Development and Innovation (Ministry of Education and Science) under grant TIC2003-04406 (Videored).

7. References

[1] ITU-T E-800. "Terms and definitions related to quality of service and network performance including dependability" Aug. 1994.

[2] N. F. Noy and D. L. McGuinness. "Ontology Development 101: A Guide to Creating Your First Ontology"'. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*, Stanford, California, USA, Mar. 2001.

[3] A. van Moorsel, "Metrics for the Internet Age: Quality of Experience and Quality of Business". *Technical Report HPL-2001-179*, Hewlet Packard Laboratories, Palo Alto, California, USA, Aug. 2001.

[4] V. Tosic, B. Esfandiari, B. Pagurek, and K. Patel, "On Requirements for Ontologies in Management of Web Services",

International Workshop on Web Services, E-Business and the Semantic Web, May 2002.

[5] Foundation for Intelligent Physical Agents “FIPA Quality of Service Ontology Specification”, Geneva, Switzerland, Nov. 2002, http://www.fipa.org/specs/fipa00094/XC00094.html.

[6] M. Tian, A. Gramm, T. Naumowicz, H. Ritter, and J. Schiller, “A Concept for QoS Integration in Web Services”, *1st Web Services Quality Workshop (WQW 2003),* Rome, Italy, Dec. 2003.

[7] C. Zhou, L. Chia, and B. Lee, “DAML-QoS Ontology for Web Services”, *Proceeding of the International Conference on Web Services 2004 (ICWS04)*, San Diego, California, USA, July 2004.

[8] E. M. Maximilien, M. P. Singh. “A Framework and Ontology for Dynamic Web Services Selection”, *IEEE Internet Computing*, 8(5), IEEE, USA, Sept. 2004, pp. 84-93.

[9] HM. Kim, A. Sengupta and J. Evermann. “MOQ: Web Services Ontologies for QOS and General Quality Evaluations”, *Proceedings of the Thirteenth European Conference on Information Systems*, Regensburg, Germany. May 2005.

[10] C. Cappiello, B. Pernici, P. Plebani, “Quality-agnostic or quality-aware semantic service descriptions”, *W3C Workshop on Frameworks for Semantics in Web Services (FSWS 2005)*, Innsbruck, Austria, June 2005.

[11] G. Dobson, R. Lock, I. Sommerville. “QoSOnt: a QoS Ontology for Service-Centric System”, *31st EUROMICRO Conference on Software Engineering and Advanced Applications*, Porto, Portugal, Aug. 2005.

[12] D.Tsesmetzis, I.Roussaki, I.Papaioannou, M.Anagnostou, “QoS awareness support in Web-Service semantics”, *International Conference on Internet and Web Applications and Services (ICIW 2006)*, Guadeloupe, French Caribbean, Feb. 2006.

[13] A. Sánchez-Macián, D. López, J. E. López de Vergara, E. Pastor, “A Framework for the Automatic Calculation of Quality of Experience in Telematic Services”. *13th Openview University Association Workshop (HP-OVUA 2006)*, Sophia Antipolis, France, May 2006.

[14] N. Oldham, K. Verma, A. Sheth, F. Hakimpour, “Semantic WS-Agreement Partner Selection”, *15th International World Wide Web Conference (WWW 2006)*, Edinburgh, Scotland, UK, May 2006.

[15] Mailing list for QoS and SLA Ontology standardization initiative https://lists.dit.upm.es/mailman/listinfo/owl-qos.

[16] H. Ludwig, A. Keller, A. Dan, R.P. King, “A Service Level Agreement Language for Dynamic Electronic Services”, *Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems (WECWIS'02)*, June, 2002.

[17] A. Andrieux, C. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, “WebServices Agreement Specification (WS-Agreement)”, June 2005.

[18] OWL-S. An OWL-based Web service ontology. http://www.daml.org/services/owl-s/1.0/

[19] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission. http://www.w3.org/Submission/SWRL/

[20] D. McGuinness, F. van Harmelen. OWL Web Ontology Language Overview. World Wide Web Consortium Recommendation http://www.w3.org/TR/owl-features/